International Journal of Innovative Research in Engineering
Volume 6, Issue 3 (May-June 2025), PP: 119-123.
www.theijire.com
ISSN No: 2582-8746

From Pixels to Patterns: A Review of Handwritten Character Recognition Methods

Pooja[1], Meenakshi Arora[2], Rohini Sharma[3]
[1]*P.G. Student, Department of CSE, Sat Kabir Institute of Technology and Management, Haryana, India.*
[2]*Assistant Professor, Department of CSE, Sat Kabir Institute of Technology and Management, Haryana, India.*
[3]*Assistant Professor, Department of computer science, GPGCW, Rohtak, Haryana, India.*

How to cite this paper:
Pooja[1], Meenakshi Arora[2], Rohini Sharma[3], "From Pixels to Patterns: A Review of Handwritten Character Recognition Methods", IJIRE-V6I3-119-123.

Abstract: *As deep learning and image processing have advanced, handwritten character recognition (HCR) has changed dramatically, moving from manually created feature extraction methods to highly flexible data-driven models. This paper offers a thorough examination of the latest HCR techniques, highlighting the interaction between deep learning architectures like Transformers, Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs) and image processing pipelines. The difficulties presented by different alphabets, writing styles, and morphological complications in scripts like Devanagari, Arabic, Chinese, and Latin are specifically addressed by cross-lingual and multi-script classification systems. The study groups current methods according to model designs, feature extraction tactics, preprocessing methods, and evaluation standards. It also draws attention to the present difficulties in attaining reliable performance on noisy datasets and low-resource languages. In order to provide information about the future direction of ubiquitous and script-agnostic HCR systems, this analysis concludes by outlining new trends and research gaps.*

Key Words: *Image processing, handwritten character recognition, Deep learning.*

I.INTRODUCTION

The goal of Handwritten Character Recognition (HCR), a basic problem in pattern recognition and computer vision, is to automatically recognize and digitize handwritten characters. Postal address interpretation, bank check verification, the digitalization of old manuscripts, and form processing systems are just a few of its many uses. Differences in handwriting styles, pen pressure, stroke width, and script-specific features contribute to the difficulty of HCR, particularly when multilingual or cross-lingual situations are taken into account. HCR systems have historically depended on statistical and rule-based techniques like Hidden Markov Models (HMMs), Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) [1]. These methods used contour-based descriptors, projections, zoning, or shape-based feature engineering [2]. Deep learning, on the other hand, has completely changed HCR by allowing models to acquire structured representations straight from raw pixel data without the need for manual feature extraction. For image-based character recognition tasks, Convolutional Neural Networks (CNNs) have emerged as the de facto standard [3], while Transformer-based models and Recurrent Neural Networks (RNNs) aid with sequential and contextual understanding [4].

Since every script, including Arabic, Devanagari, Chinese, and Latin, has distinct linguistic and structural characteristics, the cross-lingual component of HCR presents extra difficulties. To create reliable and script-agnostic recognition systems, this variability necessitates customized preprocessing, enhancement of data, and network design techniques [5]. A thorough analysis of HCR algorithms is provided in this study, with an emphasis on methods that combine deep learning, image processing, and multilingual adaptation.

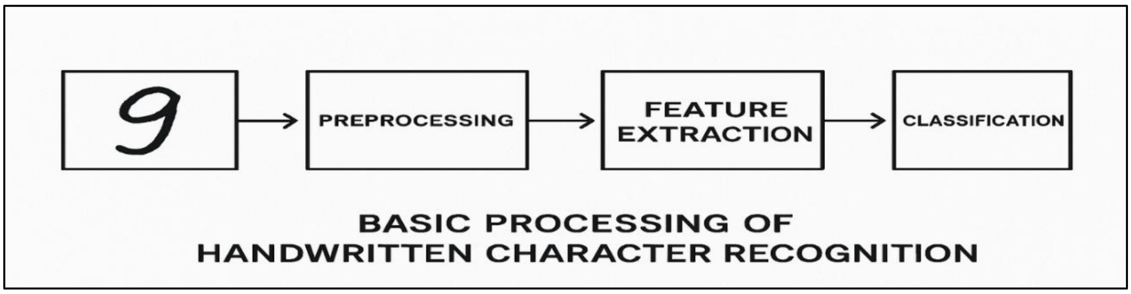

Figure 1: Basic processing of HCR

Preprocessing: Preprocessing in (HCR) refers to the set of techniques applied to raw input images to enhance their quality and prepare them for precise feature extraction and classification. It reduces noise, normalizes variations, and improves the system's robustness. Key preprocessing steps include: (i) Grayscale Conversion: Converts RGB images to grayscale to simplify further processing and lessens computational complexity. (ii) Binarization: Renovates grayscale images into binary (black & white) images using practices corresponding Otsu's thresholding. Also, it highlights the characters (foreground) and removes background. (iii) Noise Removal: Eliminates unsolicited marks, speckles, or background textures using filters (median, Gaussian, etc.). Morphological operations (e.g., erosion, dilation) are also used. (iv) Normalization: Resizes characters to a standard dimension and normalizes pixel intensities to a specific range. (v) Skew Correction: Aligns characters that are slanted or tilted using Hough Transform and projection Profile Analysis [6].

II.RESEARCH BACKGROUND

An attempt was made by Ahmad et al. [7] to use neural networks for object recognition or classification. Additionally, they proposed a three-step method for character segmentation: first, text lines are identified; second, words are identified; and third, characters are extracted from a word or subword based on complexity level and then fed into a neural network for categorization and acceptance. Character separation and obtaining from words or sub-word text lines is the main topic of the paper; word identification and the neural network used for character segmentation and identification have not yet received much attention.

Owing to the very cursive form of Urdu script, which renders the segmenting procedure challenging, Sobia et al. [8] have proposed a segmentation-free alternative in which the ligature is used whole rather than broken down into smaller sections. In their research, they fed non-segmented ligatures to an HMM recognizer after extracting global transformational features from them. They employed the Hidden Markov Model Tool Kit (HTK) to put HMM into practice. Many unique ligatures with every character are taken from a corpus-based dictionary's 5000 high-frequency terms. The text is created by examining three or more samples of each ligature.

They focused on OCR using a rule-based post-processor and HMM. The recognizer receives the ligature's primary structure, devoid of any diacritical signs, and identifies it. The accuracy of the method for printed text was 92.73%. Bhat and Hafiz [9] tried to increase the recognition ability of the HMM-based Arabic language OCR system by using a two-tier hybrid classification technique. The Arabic word (PAW) has a K-NN classifier in the second layer and an HMM-based first-tier. The performance of the proposed method is superior than that of the contemporary systems.

Among the texture descriptors are local binary patterns (LBP), local ternary patterns (LTP), and local phase quantization (LPQ). The study's evaluation metric is a comparison of the descriptors' distances from one another. Tan et al.'s work [10] proposed extracting the different geometric features using RDF datasets and ICDAR 2013, claiming a mean accuracy of 67.2%. In 2017, Akbari et al. introduced a more effective method for turning a handwritten image into a textured image with several sub-bands at different levels [11]. Every handwritten image was converted into a textured depiction, which was further divided into several wavelet sub-band levels. These subband data sequences are used to generate probabilistic finite state automata (PFSA) that produce feature vectors. ANN and SVM are trained on these features.

III.MACHINE / DEEP LEARNING TECHNIQUES

Machine Learning (ML) methods are highly useful in detecting handwritten characters because they can learn patterns from data to classify or recognize characters, even when handwriting varies across individuals, styles, and scripts. Handwritten characters have many variations (size, slant, thickness, etc.). Machine learning learns from labeled datasets to generalize these variations. It avoids the need for manually defining every possible feature. ML uses techniques to extract features like edges, curves, aspect ratios, zoning, etc. Based on features, models classify the input into one of the known character classes. ML models can still generalize even if the character is poorly written or partially missing. Robust models are trained with augmented data (rotations, distortions, etc.) to simulate real-world input.

Table1: Comparative analysis of ML based HCR Methods

Study	Model/Technique	Key Features	Dataset Used	Application/Focus	Performance Highlights
LeCun et al. [12]	LeNet-5 (CNN)	- Introduced CNN for HCR- End-to-end gradient-based learning- Pioneered document recognition	MNIST	Digit recognition (offline)	~99.2% accuracy on MNIST
Graves & Schmidhuber [13]	MDLSTM (Multidimensional LSTM)	- Handles sequential dependencies in 2D- Bidirectional recurrent connections- Suited for cursive handwriting	IAM, RIMES	Offline handwriting recognition	Outperformed HMMs, strong line-level accuracy
Shi, Bai & Yao [14]	CRNN (CNN + RNN + CTC Loss)	- End-to-end trainable- Combines spatial and sequential modeling- No	IIIT5K, SVT, ICDAR	Scene text recognition	91.9% (IIIT5K), 89.6% (SVT)

		character-level segmentation needed			
Bhunia et al. [15]	Adversarial Feature Learning	- For low-resource scripts- Domain adaptation and adversarial training- Improves recognition without large labeled data	Bangla, Devanagari, etc.	HCR in low-resource Indic scripts	Improved accuracy by ~4-5% over baselines
Sabour, Frosst & Hinton [16]	Capsule Networks	- Dynamic routing between capsules- Preserves spatial hierarchies- Robust to affine transformations	MNIST, smallNORB	Digit and object recognition	99.75% on MNIST; better generalization than CNN
Jindal & Gupta [17]	Transfer Learning + CNN	- Pre-trained CNN fine-tuned for Gurmukhi- Effective for small datasets- Simpler than full training from scratch	Gurmukhi dataset (custom)	Offline Gurmukhi character recognition	Achieved >95% accuracy using ResNet-50

IV. IMAGE PROCESSING-BASED (HCR) METHODS

Image processing plays a foundational and supportive role in the Handwritten Character Recognition (HCR) pipeline. It prepares handwritten input data for effective feature extraction and classification by enhancing image quality, reducing noise, segmenting characters, and extracting meaningful patterns. Some popular image processing methods are as follows:

Zoning Method: The image is divided into zones or blocks (e.g., 3×3, 4×4 grids). In each zone, features like pixel density or transitions are computed to form a feature vector. Captures local shape information effectively. Works well for isolated character recognition.

Projection Histograms: This technique calculates the number of foreground (ink) pixels along horizontal and vertical directions, creating histogram profiles that represent character structure. It is simple and effective for fixed-size characters.

Contour and Skeleton Features: Contours are extracted using edge detection (e.g., Canny), and skeletonization reduces characters to 1-pixel-wide strokes, preserving shape topology. It captures structure and stroke pattern.

Moment Invariants: Moments are statistical measures of shape; invariant moments are stable under rotation, translation, and scaling.

Histogram of Oriented Gradients (HOG): HOG divides the image into small cells and computes the histogram of gradient directions for edge detection and feature extraction.

Table 2: Comparison of the previous research on Image Processing based HCR methods

Study	Technique/Method	Key Features	Application in HCR	Strengths	Limitations
Freeman [18]	**Chain Code**	Encodes character contour with directional codes	Boundary representation of handwritten characters	Compact, simple encoding	Sensitive to noise and rotation
Sadjadi & Hall [19]	**Moment Invariants**	Statistical shape descriptors (e.g., Hu moments)	Rotation and scale-invariant character recognition	Effective for shape-based classification	Limited robustness to noisy or distorted images
Jain & Bhattacharjee [20]	**Gabor Filters**	Multi-scale, multi-orientation texture filters	Enhances text regions, aids segmentation	Mimics human visual processing, robust to deformation	Computationally expensive
Kimura et al. [21]	**Structural & Statistical Features**	Contour-based and projection features without segmentation	Numeral recognition (segmentation-free)	Accurate and fast	Less effective on cursive or complex scripts
Dalal & Triggs [22]	**Histogram of Oriented Gradients (HOG)**	Gradient direction	Scene text and digit recognition	Robust to small changes,	Less effective with noisy or degraded images

		histograms in image blocks		captures edge patterns	
Pradeep et al. [23]	**Diagonal Feature Extraction**	Pixel distribution along diagonals of zones	Used with neural networks for Tamil characters	Simple and effective for structured characters	Not rotation invariant, fixed grid sensitivity
Patil & Nandgaonkar (	**Vertical & Horizontal Projections**	Histograms of row/column pixel counts	Feature extraction for isolated characters	Easy to implement, efficient	Fails for cursive/connected handwriting

Table 3: Comparative analysis of some popular HCR methods (Non-ML Methods)

Method	Description	Strengths	Limitations	Best Use Cases
Template Matching	Direct comparison with stored patterns	Simple, intuitive	Not tolerant to noise, distortions, or font variation	Fixed-font printed digits/characters
Statistical Methods	Use of statistical features (zoning, moments, histograms)	Easy to implement, interpretable	Requires handcrafted features, sensitive to noise	Early HCR systems, digits
Structural/Syntactic Methods [25]	Shape, strokes, or grammar-based representations (graph, tree, etc.)	Captures structure, invariant to transformations	Complex rule design, less scalable	Online HCR, scripts with complex characters
Graph-Based Techniques [26]	Characters modeled as graphs using nodes and edges	Captures topological & geometrical info	Computationally intensive, sensitive to segmentation errors	Indic scripts, Chinese characters
Stroke-Based Analysis [27]	Focus on direction, order, and type of pen strokes	Robust to writing style variations	Requires temporal data (often online input only)	Stylus-based input, tablets
Rule-Based/Heuristic Systems [28]	Uses handcrafted rules for specific character traits	Interpretable, no training required	Brittle, lacks generalization	Simple forms (e.g., digits, checks)
Fuzzy Logic-Based Approaches [29]	Uses fuzzy sets to handle ambiguous shapes or styles	Handles uncertainty, combines well with other methods	Needs careful membership function design	Noisy or stylistically diverse inputs
Evolutionary Algorithms [30]	Optimization algorithms for feature tuning or direct recognition	Global optimization, no gradient needed	Slow convergence, computational overhead	Feature selection, hybrid systems
Hybrid Methods [31]	Combines multiple approaches (e.g., structural + statistical)	Balances accuracy and robustness	Integration complexity	Multilingual systems, noisy input
Linguistic Post-Processing [32]	Applies dictionary, grammar, or context models after recognition	Improves accuracy using context	Dependent on language data, not standalone	Word/line level OCR systems

V. CONCLUSION

The field of (HCR) has developed through a number of methodological paradigms, each of which has made a distinct contribution. The foundational layer of HCR is made up of image processing-based techniques that concentrate on segmentation, feature improvement, and noise reduction. Particularly in offline recognition tasks, these methods are crucial for preprocessing input data and obtaining significant features from unprocessed handwriting samples. By learning from data instead of depending on manually created rules, machine learning-based techniques—including conventional classifiers like SVMs, Random Forests, and k-NN—offer increased accuracy. Although they are excellent at identifying statistical patterns, they nevertheless rely significantly on carefully thought-out features and preprocessing procedures. The state-of-the-art in HCR has been further advanced by the advent of deep learning, especially convolutional neural networks (CNNs), which allow for end-to-end learning and better generalization on a variety of noisy datasets.

However, in some fields, non-machine learning-based techniques including fuzzy logic, structural analysis, stroke-based identification, and rule-based systems are still useful. These techniques provide interpretability, reduce computational expenses, and are frequently adapted to script-specific traits or limited settings (e.g., postal code reading or bank check processing). However, they typically lack the flexibility offered by data-driven techniques and suffer with substantial intra-class variability. In conclusion, image processing is still crucial for data preparation, even though machine learning and deep learning approaches right now dominate HCR due to their outstanding performance and scalability. Non-ML approaches also provide useful alternatives or supplementary strategies in applications with limited resources or rules. The most reliable and accurate HCR systems are frequently produced using a hybrid method that incorporates the advantages of these theories.

REFERENCES

1. *Plamondon, R., & Srihari, S. N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 63-84.*
2. *Trier, Ø. D., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition—A survey. Pattern Recognition, 29(4), 641-662.*
3. *Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. ICDAR.*
4. *Graves, A., & Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. NIPS.*
5. *Jindal, M., et al. (2015). Structural features for handwritten Devanagari character recognition. Journal of Intelligent Systems, 24(4), 457-471.*
6. *Balci B, Saadati D, Shiferaw D (2017) Handwritten text recognition using deep learning. In: CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring, pp 752–759.*
7. *Krishnan P, Jawahar C (2016) Matching handwritten document images. In: Proceedings of 14th European conference computer vision—ECCV Part I 2016, Amsterdam, The Netherlands, October 11–14.*
8. *Krishnan P, Jawahar C (2016) Matching handwritten document images. In: European conference on computer vision.*
9. *Jayaraman PK, Mei J, Cai J, Zheng J (2018) Quadtree convolutional neural networks. In: Proceedings of the European conference on computer vision (ECCV).*
10. *Li Y, Li Z, Qiu Q (2016) Assisting fuzzy offline handwriting recognition using recurrent belief propagation. In: 2016 IEEE symposium series on computational intelligence (SSCI).*
11. *Maitra DS, Bhattacharya U, Parui SK (2015) A CNN-based common approach to handwritten character recognition of multiple scripts. In: 2015 13th International conference on document analysis and recognition (ICDAR).*
12. *LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE.*
13. *Graves, A., & Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. NIPS.*
14. *Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. TPAMI.*
15. *Bhunia, A. K., et al. (2021). Handwriting recognition in low-resource scripts using adversarial learning. AAAI.*
16. *Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. NIPS.*
17. *Jindal, M., & Gupta, V. (2019). Transfer learning-based approach for offline handwritten Gurmukhi character recognition. Procedia Computer Science.*
18. *Freeman, H. (1961). On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computers, EC-10(2), 260–268.*
19. *Jain, A. K., & Bhattacharjee, S. (1992). Text segmentation using Gabor filters for automatic document processing. Machine Vision and Applications, 5(3), 169–184.*
20. *Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. CVPR, 886–893. (Used in OCR and character recognition too.)*
21. *Sadjadi, F. A., & Hall, E. L. (1980). Three-dimensional moment invariants. IEEE Transactions on Pattern Analysis and Machine Intelligence, (2), 127–136.*
22. *Kimura, F., Shridhar, M., & Takasawa, K. (1993). Segmentation-free handwritten numeral recognition with structural and statistical features. Pattern Recognition, 27(6), 1049–1055.*
23. *Patil, V. H., & Nandgaonkar, A. P. (2014). Character recognition using vertical and horizontal projection feature with neural network. IJERT, 3(2), 1816–1819.*
24. *Pradeep, J., Srinivasan, E., & Himavathi, S. (2011). Diagonal based feature extraction for handwritten character recognition system using neural network. International Journal of Computer Applications, 8(9), 35–40.*
25. *Nagy, G. (1980). At the frontiers of OCR. Proceedings of the IEEE.*
26. *Liu, C.-L., Nakashima, K., Sako, H., & Fujisawa, H. (2004). Handwritten digit recognition: Benchmarking of state-of-the-art techniques. Pattern Recognition.*
27. *Plamondon, R., & Srihari, S. N. (2000). Online and offline handwriting recognition: A comprehensive survey. IEEE T-PAMI.*
28. *Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The state of the art in online handwriting recognition. IEEE T-PAMI.*
29. *Dutta, A., & Chaudhuri, B. B. (1993). A fuzzy approach to handwritten Devanagari character recognition. Proceedings of the 2nd Int. Conf. on Document Analysis and Recognition (ICDAR).*
30. *Haupt, R. L., & Haupt, S. E. (2004). Practical Genetic Algorithms. Wiley.*
31. *Kim, G., & Kim, V. (2000). Hierarchical decision making for handwritten character recognition: Combination of global and local decisions. Pattern Recognition Letters.*
32. *Srihari, S. N. (1993). Use of lexicons in recognizing handwritten words. Document Recognition II, SPIE.*